

PATENT APPLICATION

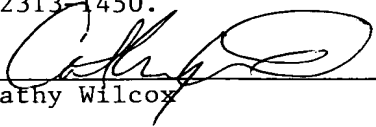
of

Ossi Kalevo

for a

CAMERA OUTPUT FORMAT FOR REAL TIME VIEWFINDER/VIDEO
IMAGE

I hereby certify that this communication is being deposited with the United States Postal Service on the date shown below in an envelope marked as, and with sufficient postage as, "Express Mail - Post Office to Addressee," Mailing Label # EV303711555US, addressed to Director for Patents, Mail Stop Patent Application, Alexandria, VA 22313-1450.


Cathy Wilcox

10.23.03

Date

CAMERA OUTPUT FORMAT FOR REAL TIME
VIEWFINDER/VIDEO IMAGE

Field of the Invention

5 This invention generally relates to formatting of a real-time video image in electronic devices and more specifically to horizontal and vertical downscaling of the real-time video image using a camera sensor and a processing block.

Background of the Invention

10 1. Field of technology and background

 A camera sensor is used when still image or video are captured. The output image of the camera sensor can use different image formats e.g. RGB8:8:8, RGB5:6:5, YUV4:2:2, YUV4:2:0, or raw-Bayer image. When the image is displayed on a viewfinder (VF), which usually has a smaller resolution than the camera sensor,
15 the image has to be read from the camera sensor and down-sampled (or downscaled) to the display resolution. When the video image is encoded the resolution of the video frame is also usually smaller than the sensor resolution. Therefore, some kind of downscaling is required. So far the problem of downscaling of a high-resolution image to a low-resolution image was solved by a number of methods which are listed
20 below:

 1) Sending the high-resolution result image using e.g. YUV4:2:2 or RGB8:8:8 image format to the processing unit and doing processing there;

 2) Down-sampling the high-resolution result image in camera sensor's hardware to the low-resolution result image and then sending the low-resolution result
25 image using e.g. YUV4:2:0 or RGB5:6:5 image format to processing unit (this kind of solution is used e.g. in 7650 and 3650 Nokia mobile phones);

 3) Sending the sensor's high-resolution raw-Bayer image to the processing unit and doing all the required processing there.

The above downscaling solutions require a significant amount of memory, processing power and bus capability and further improvements are needed.

Summary of the Invention

5 The object of the present invention is to provide a simplified methodology for downscaling of the real-time video image using a camera sensor and a processing block.

According to a first aspect of the present invention, a method for generating a real-time vertically and horizontally downscaled video signal of a video image by an image generating and processing block comprises the steps of: generating a real-time
10 video signal of the video image by a camera sensor of the image generating and processing block; generating a real-time horizontally downscaled video signal using horizontal downscaling of the real-time video signal by the camera sensor; and generating the real-time vertically and horizontally downscaled video signal using vertical downscaling of the real-time horizontally downscaled video signal by a
15 processing block of the image generating and processing block. Further, the camera sensor may have a camera memory. Still further, the processing block may have a processing memory.

In further accord with the first aspect of the invention, before the step of generating a real-time vertically and horizontally downscaled video signal, the
20 method further comprises the step of providing said real-time horizontally downscaled video signal from the camera sensor to the processing block through a camera compact port (CCP) bus of the image generating and processing block.

Still further according to the first aspect of the invention, the method further comprises the step of providing the real-time vertically and horizontally downscaled
25 video signal indicative of the video image through an internal bus to a real-time viewfinder display and displaying said video image on the real-time viewfinder display. Further, the image generating and processing block may be a part of a camera-phone mobile device. Still further, the processing block may be a base band (BB) engine of the camera-phone mobile device. Yet still further, the method may

further comprise the steps of: encoding the real-time vertically and horizontally
downscaled video signal by a video packing block of the image generating and
processing block, thus generating an encoded video signal; and providing said
encoded video signal through a further internal bus optionally to a file/stream block
5 and to a phone memory of the camera-phone mobile device.

Further still according to the first aspect of the invention, the method may
further comprise the step of encoding the vertically and horizontally downscaled
video signal by a video packing block of the image generating and processing block,
thus generating an encoded video signal.

10 According to a second aspect of the invention, an image generating and
processing block comprises: a camera sensor, responsive to a video image, for
generating a real-time video signal of the video image and for further generating a
real-time horizontally downscaled video signal using horizontal downscaling of the
real-time video signal by the camera sensor; and a processing block, responsive to the
15 real-time horizontally downscaled video signal, for generating a real-time vertically
and horizontally downscaled video signal using vertical downscaling of the real-time
horizontally downscaled video signal. Further, the camera sensor may have a camera
memory. Still further, the processing block may have a processing memory.

According further to the second aspect of the invention, the image generating
20 and processing block further comprises a camera compact port (CCP) bus, responsive
to the real-time horizontally downscaled video signal from the camera sensor, for
providing the real-time horizontally downscaled video signal to the processing block.

Further according to the second aspect of the invention, the receiving terminal
is further responsive to a software request command by a user, provides a message
25 retrieval request signal containing a terminal signal indicative of a terminal
information and optionally a multipurpose internet mail extensions (MIME) signal
indicative of a terminal-specific MIME information, provides a software request
signal to an Internet server, provides a URL image signal to the user, and renders the
further multimedia message signal indicative of the multimedia message perceptible

by the user. Also, the receiving terminal may be responsive to a message notification signal.

Still further according to the second aspect of the invention, the system further comprises a sending terminal, for providing a multimedia message signal to the
5 multimedia messaging service center.

According to a third aspect of the invention, a camera-phone mobile device, comprises: an image generating and processing block for generating a real-time vertically and horizontally downsampled video signal of a video image, and for encoding said real-time vertically and horizontally downsampled video signal thus
10 generating an encoded video signal; and a real-time viewfinder display, responsive to the real-time vertically and horizontally downsampled video signal, for providing a display of the video image indicative by said real-time vertically and horizontally downsampled video signal.

In further accordance with the third aspect of the invention, a camera-phone
15 mobile device, may further comprise: a file/stream block, responsive to the encoded signal, for providing a call connection to other mobile devices; and a phone memory, responsive to the encoded signal, for providing the encoded signal.

Yet further still according to the third aspect of the invention, the image generating and processing block of the camera-phone mobile device comprises: a
20 camera sensor, responsive to the video image, for generating the real-time video signal of the video image and for further generating a real-time horizontally downsampled video signal using horizontal downscaling of the real-time video signal by the camera sensor; and a processing block, responsive to the real-time horizontally downsampled video signal, for generating the real-time vertically and horizontally
25 downsampled video signal using vertical downscaling of the real-time horizontally downsampled video signal. Further, the processing block may be a base band (BB) engine of the camera-phone mobile device. Still further, the camera sensor may have a camera memory. Also further, the processing block may have a processing memory. Also still further, the camera-phone mobile device may further comprise a camera
30 compact port (CCP) bus, responsive to the real-time horizontally downsampled video

signal from the camera sensor, for providing the real-time horizontally downsampled video signal to the processing block.

5 This invented image format offers the possibility to process the image as much as possible in the camera sensor without additional memory and using as little processing power as possible. Also the amount of the transmitted image information is reduced as much as possible and at the same time the phase of the color components in the same line are equalized. This phase equalization and data reduction offer the high-quality result image processing with minimal processing power requirements in the processing block. The invented image format enables the
10 reduction of memory, processing power and bus capability, when the smaller resolution output image is needed to process from the higher resolution input image.

Brief Description of the Drawings

For a better understanding of the nature and objects of the present invention, reference is made to the following detailed description taken in conjunction with the
15 following drawings, in which:

Figure 1 is a block diagram representing an example of a camera-phone mobile device utilizing an image generating and processing block for generating and processing a real-time vertically and horizontally downsampled video signal, according to the present invention.

20 Figure 2 shows a flow chart for generating and processing of a real-time vertically and horizontally downsampled video signal, according to the present invention.

Figure 3 shows an example of a raw-Bayer nxm video image generated by a camera sensor and used for downscaling, according to the present invention.

25 Figure 4 shows an example of a horizontally downsampled nxM video image generated using horizontal downscaling of a real-time video image signal by a camera sensor, according to the present invention.

Figure 5 shows an example of a vertically and horizontally downsampled NxM video image generated using vertical downscaling of a horizontally downsampled nxM video image by a processing block, according to the present invention.

Best Mode for Carrying Out the Invention

5 The present invention provides a methodology for generating a real-time vertically and horizontally downsampled video signal of a real-time video signal generated by a camera sensor by, first, generating a real-time horizontally downsampled video signal by said camera sensor, second, providing said real-time horizontally downsampled video signal to the processing block through a bus (e.g., compact camera
10 port, CCP, bus) and, third, generating the real-time vertical downscaling of said horizontally downsampled video signal by said processing block.

Figure 1 is a block diagram representing only one example among many others of a camera-phone mobile device 10 utilizing an image generating and processing block 12 for generating and processing a real-time vertically and
15 horizontally downsampled video signal, according to the present invention.

The camera sensor 14 of the image generating and processing block 12 generates a real-time video signal in response to a video image 11. Said real-time video signal is horizontally downsampled by said camera sensor 14 requiring insignificant processing capabilities. A small capacity camera memory 14a within the
20 camera sensor 14 is used for assisting of said horizontal downscaling performed by the camera sensor 14, which produces a horizontally downsampled video signal 18. An algorithm of horizontal downscaling is discussed below in details regarding Figure 4. Then said horizontally downsampled video signal 18 from the camera sensor 14 is provided to a processing block 16 of the image generating and processing block 12
25 through a compact camera port (CCP) bus 15.

The processing block 16 generates a real-time vertically and horizontally downsampled video signal 20 using vertical downscaling of the real-time horizontally downsampled video signal 18. A small capacity processing memory 16a within the processing block 16 is used for assisting of said vertical downscaling performed by
30 the processing block 16. An algorithm of the vertical downscaling is discussed below

in details regarding Figure 5. For the example of the camera-phone mobile device **10** presented in Figure 1, the processing block **16** is practically a base band engine which performs statistic collection for color correction, correction of colors, and possible other processes related to image quality in addition to said vertical downscaling.

5 The real-time vertically and horizontally downsampled video signal **20** indicative of the video image **11** is provided (optionally) through an internal bus **25a** to a real-time viewfinder display **22** of the camera-phone mobile device **10** for displaying said video image **11** on said real-time viewfinder display **22**. The real-time vertically and horizontally downsampled video signal **20** is also provided (again
10 optionally) through an internal bus **25b** to a video packing block **24** of the image generating and processing block **24**, said video packing block **24** generates an encoded video signal **27** from the real-time vertically and horizontally downsampled video signal **20**. The video packing block **24** can be a part of the base band engine **16**, according to the present invention. Finally the encoded video signal **27** is provided to
15 a phone memory **28a** and/or to a file/stream block **28** of the camera-phone mobile device **10** through further internal buses **27a**, **27b** and **27c**, respectively, and possibly sent to other phone through a phone connection **28b**.

 This invention can optionally require also bit companding when the image information is transferred between the camera sensor **14** and processing unit **16**. This
20 companding can be done using e.g. gamma based look-up-table (LUT). Even better companding solution can be utilizing the method described in invention "High Quality Codec from N bits to M Bits Using Fixed Length Codes" by Kalevo, submitted as patent application in Finland (FI application number 20030045, filed on 1/13/2003). Said companding, if necessary, is implemented by the camera sensor **14**
25 and uncomparing is performed by the processing block **16**, respectively.

 There are a number of possible variations of the application shown in Figure 1. For example, when the actual still image is processed, Nokia Image Processing System (NIPS) can be used by the processing block **16**. Additionally, for video and viewfinder images some auto white balance (AWB) solution, e.g., described in
30 invention "Automatic Color Balance Adjustment for Digital Cameras" by Kalevo et al., submitted as patent application in Finland (FI application number 20035162, filed

on 9/24/2003) can be used as well. It is also possible to do specific image corrections only for the image to be shown on the real-time viewfinder display **22** at the same time as color correction in the processing block **16**, but not for the image, which is entering the video coding in the video packing block **24**. The vertical downscaling can also be performed by the camera sensor **14**, such that the camera sensor output can be a horizontally and vertically downsampled signal but it is still a raw image without any corrections. Finally, it is noted that the horizontally and vertically downsampled image can also be sent using 4-color component values at the same pixel location instead of 3-color component values shown in Figure 5 and described below. This 4-color component values approach assumes, for example, sending two G-components (G in red line and G in blue line) by the camera sensor **14** separately and combining them later in the processing block **16**.

Figure 2 shows a flow chart for generating and processing of a real-time vertically and horizontally downsampled video signal according to the present invention. The flow chart of Figure 2 only represents one possible scenario among others. In a method according to the present invention, in a first step **30**, the real-time video signal of the video image **11** is generated by the camera sensor **14**. In a next step **32**, the horizontally downsampled video signal **18** is generated by the camera sensor **14** with optional companding as mentioned above. In a next step **34**, horizontally downsampled video signal **18** is provided to the processing block **16** (e.g., BB engine of the camera-phone mobile device). In a next step **36**, the horizontally and vertically downsampled video signal **20** is generated by the processing block **16** (e.g., BB engine) by vertical downscaling of the horizontally downsampled video signal **18**.

In a next step **38**, the horizontally and vertically downsampled video signal **20** is provided to the real-time viewfinder display **22** and/or to the video packing block **24**. In a next step **40**, the horizontally and vertically downsampled video signal **20** is displayed (optionally) on the real-time viewfinder display **22**. In a next step **42**, the horizontally and vertically downsampled video signal **20** is encoded (optionally) by the video packing block **24**, forming the encoded video signal **27**. Finally, in a next step **44**, the encoded video signal **27** is provided to the phone memory **28a** and/or to the file/stream block **28** for call connection to other mobile devices.

Figures 3-5 demonstrate an example among many others of how the VGA-resolution raw-Bayer image generated by the camera sensor **14** can be processed to the QQVGA-resolution image format, according to the present invention. In particular, Figure 3 shows an example of a VGA-resolution raw-Bayer $n \times m$ (number of columns $m=640$ and number of lines $n=480$) video image generated by the camera sensor **14** and used for downscaling by said camera sensor **14**, wherein g , r and b correspond to standard color components (green, red and blue) with indexes in parentheses identifying a line number (from 0 to $n-1$) and a column number (from 0 to $m-1$).

Figure 4 shows an example of a horizontally downsampled $n \times M$ video image generated using horizontal downscaling of a real-time video image signal by a camera sensor **14**, according to the present invention, wherein G , R and B corresponds to standard color components (green, red and blue) with indexes in parentheses identifying the line number (from 0 to $n-1$) and a downsampled column number (from 0 to $M-1$).

Image of Figure 4 consists of even and odd numbered lines. For each line only two color component pixel values are available from the three available color component pixel values of the original video image of Figure 3. For example, the even lines consist of R and G color component pixel values and the odd lines consist on B and G color component pixel values. Two color component values in the same line are in the same phase. This means that for each pixel position there are two colour component pixel values available. E.g. even line pixels can be presented as $R(0,0)$, $G(0,0)$, $R(0,1)$, $G(0,1)$, $R(0,2)$, $G(0,2)$, ..., $R(0,M-1)$ and $G(0,M-1)$, where M is the number of the pixels in line. So the total amount of the pixel components in the line is $2 \times M$. This M value is typically 128 (sub-QCIF), 160 (QQVGA), 176 (QCIF), 320 (QVGA) or 352 (CIF). Also the pixel component phases in odd and even lines are the same in horizontal direction. This means that, for example, the $B(1,0)$ and $G(1,0)$ pixels in the 1st ($n=1$) line are the same as the $R(0,0)$, $G(0,0)$, $R(2,0)$ and $G(2,0)$, pixels in 0th ($n=0$) and 2nd ($n=2$) lines, respectively. It is noted, that according to the present invention, when the pixels in the lines (horizontally downsampled video signal **18**) are transmitted through the information channel (CCP bus **15**) the order of the

pixels in the same position has to be specified: when the even line is transmitted, the order of pixels, e.g. for the 0th line, is G(0,0), R(0,0), G(0,1), R(0,1), etc., and when the odd line is transmitted the order of pixels, e.g. for the 1st line, is B(1,0), G(1,0), B(1,1), G(1,1) etc.

5 For the QQVGA-resolution image format, the image ratio is $\text{Min}(m/M, n/N) = \text{Min}(640/160, 480/120) = 4$. This means that one pixel in invented image format in horizontal or vertical direction corresponds to the distance in raw-Bayer image of 4 pixels. The optimal first position, which utilizes the whole image information in horizontal direction in Figure 4, is between pixels 1 and 2 (1.5). This means that the
10 output pixel values can be calculated as follows:

Even lines:

$$G(0,0) = (3g(0,0) + 4g(0,2) + g(0,4))/8,$$

$$R(0,0) = (r(0,-1) + 4r(0,1) + 3r(0,3))/8,$$

$$G(0,1) = (3g(0,4) + 4g(0,6) + g(0,8))/8,$$

15 $R(0,1) = (r(0,3) + 4r(0,5) + 3r(0,7))/8, \text{ etc.};$

Odd lines:

$$B(1,0) = (3b(1,0) + 4b(1,2) + b(1,4))/8,$$

$$G(1,0) = (g(1,-1) + 4g(1,1) + 3g(1,3))/8,$$

$$B(1,1) = (3b(4) + 4b(6) + b(8))/8,$$

20 $G(1,1) = (g(1,3) + 4g(1,5) + 3g(1,7))/8, \text{ etc.}$

The phase of the pixels can be selected differently. E.g., if the computational complexity is needed to be minimized, the position of the first pixel can be selected to be (0,1). This means that the output values can be calculated as follows:

Even lines:

25 $G(0,0) = (g(0,0) + g(0,2))/2,$

$$R(0,0) = (r(0,-1) + 2r(0,1) + r(0,3))/4,$$

$$G(0,1) = (g(0,4) + g(0,6))/2,$$

$$R(0,1) = (r(0,3) + 2r(0,5) + r(0,7))/4, \text{ etc.};$$

Odd lines:

$$B(1,0) = (b(1,0) + b(1,2))/2,$$

$$G(1,0) = (g(1,-1) + 2g(1,1) + g(1,3))/4,$$

$$5 \quad B(1,1) = (b(4) + b(6))/2,$$

$$G(1,1) = (g(1,3) + 2g(1,5) + g(1,7))/4, \text{ etc.}$$

The values of $r(0,-1)$ and $g(1,-1)$ and other pixel values with negative indexes correspond to the red and green component values of the pixels lying on the left side of the pixels $(0,0)$, $(1,0)$, ..., $(n-1,0)$, respectively. If there is no value for these
 10 marginal pixels, the value for pixels $(0,0)$, $(1,0)$, ..., $(n-1,0)$ or $(0,1)$, $(1,1)$, ..., $(n-1,1)$ depending on the component's color can be used instead, respectively.

The horizontal downscaling described above is done in camera sensor **14** and it does not require line memories. After downscaling, the horizontally downscaled video signal **18** is transmitted to the processing block **16** through the CCP bus **15** as
 15 described above and the vertical downscaling is done in the processing block **16**. The result image format is shown in Figure 5. Thus Figure 5 shows an example of a vertically and horizontally downscaled $N \times M$ video image (e.g., RGB8:8:8 format) generated using vertical downscaling of a horizontally downscaled $n \times m$ video image of Figure 4 by a processing block **16**, according to the present invention.

20 Again, for the image format for $N \times M$ image (RGB8:8:8) of QQVGA-resolution image format, the optimal first position, which utilises the whole image information in vertical direction, is between pixels 1 and 2. This means that the output pixel values can be calculated as follows:

$$R'(0,0) = (3R(0,0) + 4R(2,0) + R(4,0))/8,$$

$$25 \quad G'(0,0) = (G(-1,0) + 3G(0,0) + 4G(1,0) + 4G(2,0) + 3G(3,0) + G(4,0))/16,$$

$$B'(0,0) = (B(-1,0) + 4B(1,0) + 3B(3,0))/8$$

$$R'(1,0) = (3R(4,0) + 4R(6,0) + R(8,0))/8,$$

$$G'(1,0) = (G(3,0) + 3G(4,0) + 4G(5,0) + 4G(6,0) + 3G(7,0) + G(8,0))/16,$$

$$B'(1,0) = (B(3,0) + 4B(5,0) + 3B(7,0))/8, \text{ etc.}$$

The phase of the pixels can be selected differently. E.g., if the computational complexity is needed to be minimized, the position of the first pixel can be selected to be (1,0). This means that the output values can be calculated as follows:

$$5 \quad R'(0,0) = (R(0,0) + R(2,0))/2,$$

$$G'(0,0) = (G(-1,0) + 2G(0,0) + 2G(1,0) + 2G(2,0) + G(3,0))/8,$$

$$B'(0,0) = (B(-1,0) + 2B(1,0) + B(3,0))/4$$

$$R'(1,0) = (R(4,0) + R(6,0))/2,$$

$$G'(1,0) = (G(3,0) + 2G(4,0) + 2G(5,0) + 2G(6,0) + G(7,0))/8,$$

$$10 \quad B'(1,0) = (B(3,0) + 2B(5,0) + B(7,0))/4, \text{ etc.}$$

The values of; $G(-1,0)$, $B(-1,0)$ and other pixel values with negative indexes correspond to the green and blue component values of the pixels lying on the upper side of the pixels $(0,0)$, $(0,1)$, ..., $(0, M-1)$, respectively. If there is no value for this marginal pixels, the value for pixel $(0,0)$, $(0,1)$, ..., $(0, M-1)$ or $(1,0)$, $(1,1)$, ..., $(1, M-1)$ depending on the component's color can be used instead, respectively.

When the amount of pixels is correct and all the pixels are containing all the color component values, minimally only the AWB (with linearization, vignetting elimination, color gamut conversion and gamma mapping) is additionally required when the high quality result image is processed.

20 The situation is slightly more complicated when e.g. the VGA raw-Bayer image is used for the QCIF-video. In this scenario the image ratio $(\text{Min}(640/176, 480/144) = \text{Min}(40/11, 10/3) = 10/3)$ is not an integer value. Also it is obvious that the image (aspect) ratios in horizontal and vertical directions are not equal and, therefore, part of the image is discarded when the downscaling is done as further explained
25 below. In this case the set of pixels on the left and right sides of the image are not used.

It is noticed that in this example it is desirable to keep the aspect ratio equal so the scaling ratio for the horizontal and vertical directions is the same. In some cases the aspect ratio can also be changed (e.g., full image information is used) or have to

be changed (e.g., from a progressive image sensor the interlaced image is needed to be used for displaying) and so the scaling ratio for the horizontal and vertical directions can be different as well. Also the cropping of the image can be made before the downscaling as in this example or after said downscaling. The amount of the
 5 cropped pixels can also be much higher for both directions and the image zooming for the viewfinder or video image can be applied by changing the scaling ratio as well.

Thus in this case the amount of the used pixels is $M \cdot n / N = 586.66$. Then the first pixel horizontal position is optimally placed in the raw-Bayer image position ($m - M \cdot n / N / 2 + (n / N) / 2 - 1 / 2 = 27.833$). Also the processing window moves such that
 10 after 3 processed output pixel values 10-pixel shift in the input image is performed. In the example below the horizontal downscaling from VGA to QCIF is done and the first pixel position is selected to be 27.666. Then the downscaled output pixel values can be calculated as follows:

Even lines:

$$15 \quad G(0,0) = (3g(0,26) + 6g(0,28) + g(0,30)) / 10,$$

$$R(0,0) = (6r(0,27) + 4r(0,29)) / 10,$$

$$G(0,1) = (5g(0,30) + 5g(0,32)) / 10,$$

$$R(0,1) = (2r(0,29) + 6r(0,31) + 2r(0,33)) / 10,$$

$$G(0,2) = (g(0,32) + 6g(0,34) + 3g(30,6)) / 10,$$

$$20 \quad R(0,2) = (4r(0,33) + 6r(0,35)) / 10,$$

$$G(0,3) = (3g(0,36) + 6g(0,38) + g(0,40)) / 10,$$

$$R(0,3) = (6r(0,37) + 4r(0,39)) / 10, \text{ etc.}$$

Odd lines:

$$B(1,0) = (3b(1,26) + 6b(1,28) + b(1,30)) / 10,$$

$$25 \quad G(1,0) = (6g(1,27) + 4g(1,29)) / 10,$$

$$B(1,1) = (5b(1,30) + 5b(1,32)) / 10,$$

$$G(1,1) = (2g(1,29) + 6g(1,31) + 2g(1,33)) / 10,$$

$$B(1,2) = (b(1,32) + 6b(1,34) + 3b(1,36)),$$

$$G(1,2) = (4g(1,33) + 6g(1,35)) / 10,$$

$$B(1,3) = (3b(1,36) + 6b(1,38) + b(1,40)),$$

$$G(1,3) = (6g(1,37) + 4g(1,39)) / 10, \text{ etc.}$$

- 5 Another example is how one mega-pixel image (1152x864) can be
downscaled horizontally to the QQVGA image. In this scenario the image ratio is 7.2
(1152/160 = 864/120 = 36/5), and the first pixel position selected to be optimal is 3.1
((864/120)/2 – 1/2). In this case after 5 processed output pixel values 36-pixel shift in
input image is performed. This example shows only the processing of even line pixels
10 as follows:

$$G(0,0) = (15g(0,0) + 20g(0,2) + 20g(0,4) + 17g(0,6)) / 72$$

$$R(0,0) = (5r(0,-1) + 20r(0,1) + 20r(0,3) + 20*0r(0,5) + 7r(0,7)) / 72$$

$$G(0,1) = (3g(0,6) + 20g(0,8) + 20g(0,10) + 20g(0,12) + 9g(0,14)) / 72$$

$$R(0,1) = (13r(0,7) + 20r(0,9) + 20r(0,11) + 19r(0,13)) / 72$$

$$15 \quad G(0,2) = (11g(0,14) + 20g(0,16) + 20g(0,18) + 20g(0,20) + 1g(0,22)) / 72$$

$$R(0,2) = (r(0,13) + 20r(0,15) + 20r(0,17) + 20r(0,19) + 11r(0,21)) / 72$$

$$G(0,3) = (19g(0,22) + 20g(0,24) + 20g(0,26) + 13g(0,28)) / 72$$

$$R(0,3) = (9r(0,21) + 20r(0,23) + 20r(0,25) + 20r(0,27) + 3r(0,29)) / 72$$

$$G(0,4) = (7g(0,28) + 20g(0,30) + 20g(0,32) + 20g(0,34) + 5g(0,36)) / 72$$

$$20 \quad R(0,4) = (17r(0,29) + 20r(0,31) + 20r(0,33) + 15r(0,35)) / 72$$

$$G(0,5) = (15g(0,36) + 20g(0,38) + 20g(0,40) + 17g(0,42)) / 72$$

$$R(0,5) = (5r(0,35) + 20r(0,37) + 20r(0,39) + 20r(0,41) + 7r(0,43)) / 72, \text{ etc.}$$

The vertical downscaling for the above two scenarios is performed in a similar manner as for the horizontal downscaling.